KATHMANDU UNIVERSITY

Department of Computer Science and Engineering

DHULOKHEL, KAVRE

COURSE CODE: (COMP342)

LAB REPORT: 2

Submitted by: Prithivi Raj Shrestha

Submitted to: Mr. Dhiraj Shrestha

Roll no: 50

Group: C.E. (3rd year, 2nd semester)

Date of Submission: 18th August, 2020

1. **Digital differential line drawing algorithm**

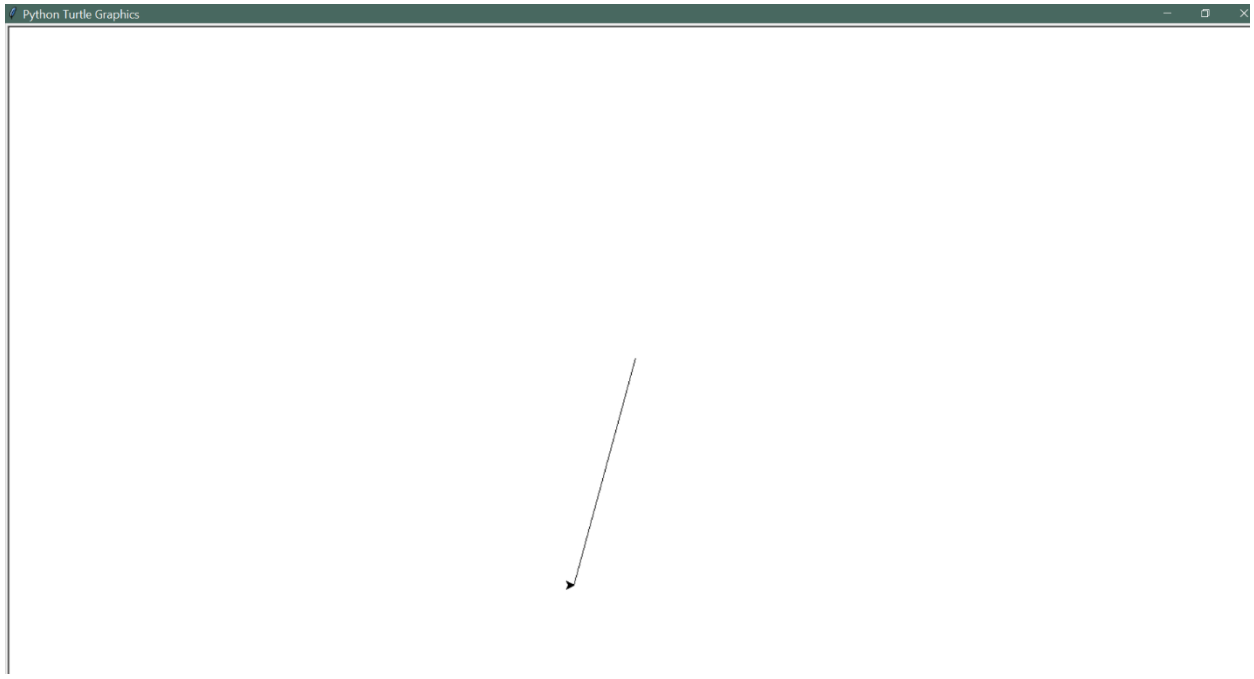
Step 1: Start Algorithm
Step 2: Declare x1, y1, x2, y2, dx, dy, x, y.
Step 3: Enter values of x1, x2, y1, y2.
Step 4: Calculate dx = x2-x1 and dy = y2-y1
Step 5: If ABS (dx) > ABS (dy)
Then step = abs (dx)
Else step = abs(dy)
Step 6: xinc=dx/step
yinc=dy/step
assign x = x1
assign y = y1
Step 7: Set pixel (x, y)
Step 8: x = x + xinc
y = y + yinc
Set pixels (Round (x), Round (y))
Step 9: Repeat step 9 until x = x2
Step 10: End Algorithm

Source Code:

```
import turtle

DDA = turtle.Turtle()


def line(x1,y1,x2,y2):
    x = x1
    y = y1
    DDA.penup()
    DDA.goto(x,y)
    DDA.pendown()
    i = 0
    dx = x2-x1
    dy = y2-y1
    if abs(dx) > abs(dy):
        steps = abs(dx)

    else:
        steps = abs(dy)
```

```
    x_incerement = dx/float(steps)
    y_incerement = dy/float(steps)

    for i in range(steps):
        x = x+ x_incerement
        y = y+ y_incerement
        print(round(x),round(y))

        DDA.goto(x,y)
line(-3,1,-78,-275)


turtle.done()
```

Outout:



2. Bresenham Line Drawing Algorithm for both slopes(|m|<1 and |m|>=1)

Algorithm:

Step 1: Input end points as (x1,y1) and (x2,y2)
Step 2: Plot first point(x1,y1)
Step 3: Calculate all required variables
Step 4: if |m|<1 go from 5 to 7 else go from 8 to 10
Step5: Calculate the initial decision parameter as,
D0 = 2dy-dx
Step 6: for each xk starting at k = 0
If dk<0 choose the next point as (xk+1,yk) and d(k+1) = dk+2dy
Else, choose next point as (xk+1,yk+1)
Step 7: Repeat step 5 and 6 for dx times

Step8: Calculate the initial decision parameter as,
D0 = 2dx-dy
Step 9: for each yk starting at k = 0
If dk<0 choose the next point as (xk,yk+2) and d(k+1) = dk+2dx
Else, choose next point as (xk+1,yk+1)
Step 10: Repeat step 8 and 9 for dy times

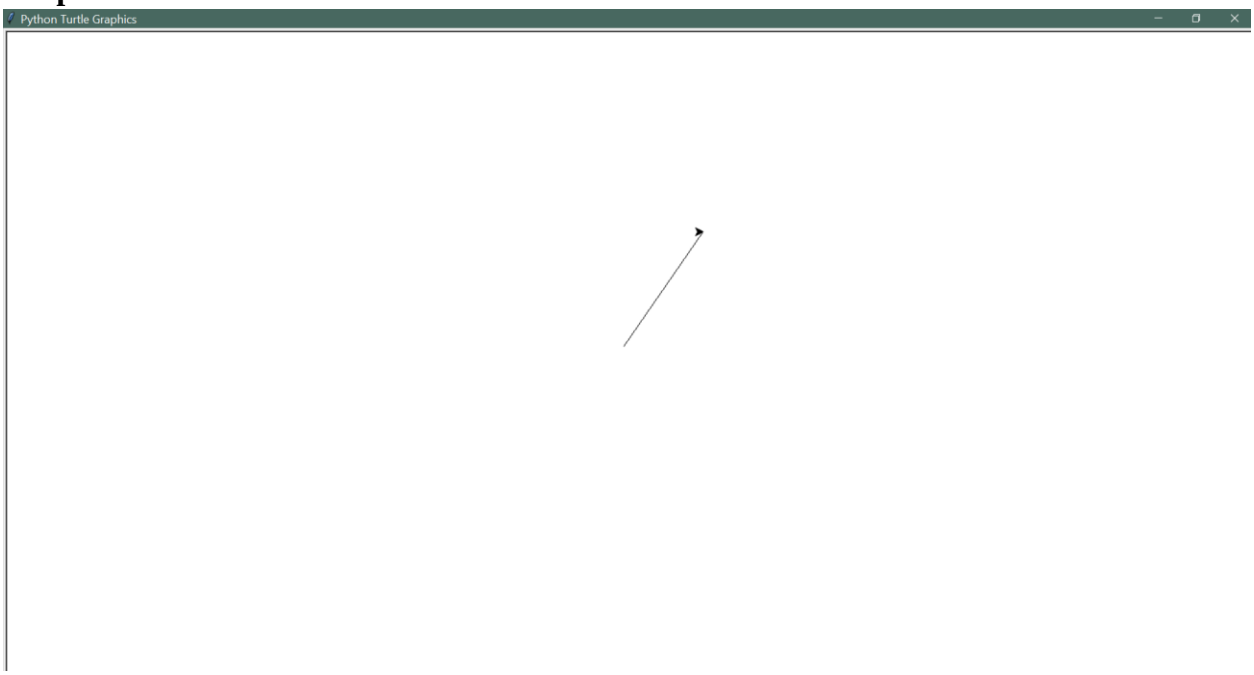
Source cose:

```
import turtle

BLA = turtle.Turtle()

def line(x1,y1,x2,y2):

   xs = 0
   ys = 0
   if x1<x2:
      xs = 1

   else:
      xs = -1

   if y1<y2:
      ys = 1

   else:
      ys = -1
```

```
x = x1
y = y1
BLA.penup()
BLA.goto(x,y)
BLA.pendown()
dx = x2-x1
dy = y2-y1

if abs(dy)<abs(dx):
    p0 = 2*dy-dx

    while abs(x) < abs(x2):
        if p0<0:
            x = x+1*xs
            BLA.goto(x,y)
            p0 = p0+2*dy
            print(x,y)

        else:
            x = x+1*xs
            y = y+1*ys
            p0 = p0+2*dy-2*dx
            BLA.goto(x,y)
            print(x,y)


elif abs(dx)<abs(dy):
    p0 = 2*dx-dy

    while abs(y)<abs(y2):
        if p0<0:
            print(xs)

            y = y+1*ys
            BLA.goto(x,y)
            p0 = p0+2*dx
            print(x,y)


        else:
            x = x+1*xs
            y = y+1*ys
```

```
        p0 = p0+2*dx-2*dy
        BLA.goto(x,y)
        print(x,y)



line(-10,20,87,160)
turtle.done()
```

Output:



3. Mid-point line drawing algorithm

Algorithm:

Step 1: Input end points as (x1,y1) and (x2,y2)
Step 2: Plot first point(x1,y1)
Step 3: Calculate all required variables
Step 4: if |m|<1 go from 5 to 7 else go from 8 to 10
Step5: Calculate the initial decision parameter as,
D0 = 2dy-dx
Step 6: for each xk starting at k = 0
If dk<0 choose the next point as (xk+1,yk) and d(k+1) = dk+2dy
Else, choose next point as (xk+1,yk+1)
Step 7: Repeat step 5 and 6 for dx times

Step8: Calculate the initial decision parameter as,
D0 = 2dx-dy
Step 9: for each yk starting at k = 0
If dk<0 choose the next point as (xk,yk+2) and d(k+1) = dk+2dx
Else, choose next point as (xk+1,yk+1)
Step 10: Repeat step 8 and 9 for dy times

Source cose:

```
import turtle

MLP = turtle.Turtle()

def line(x1,y1,x2,y2):

   xs = 0
   ys = 0
   if x1<x2:
      xs = 1

   else:
      xs = -1

   if y1<y2:
      ys = 1

   else:
      ys = -1
```

```
x = x1
y = y1
MLP.penup()
MLP.goto(x,y)
MLP.pendown()
dx = x2-x1
dy = y2-y1

if abs(dy)<abs(dx):
    p0 = 2*dy-dx

    while abs(x) < abs(x2):
        if p0<0:
            x = x+1*xs
            MLP.goto(x,y)
            p0 = p0+2*dy
            print(x,y)

        else:
            x = x+1*xs
            y = y+1*ys
            p0 = p0+2*dy-2*dx
            MLP.goto(x,y)
            print(x,y)


elif abs(dx)<abs(dy):
    p0 = 2*dx-dy

    while abs(y)<abs(y2):
        if p0<0:
            print(xs)

            y = y+1*ys
            MLP.goto(x,y)
            p0 = p0+2*dx
            print(x,y)


        else:
            x = x+1*xs
            y = y+1*ys
```

```
        p0 = p0+2*dx-2*dy
        MLP.goto(x,y)
        print(x,y)



line(-10,20,87,160)
turtle.done()
```

Output: